Weather Forecast

Welcome to Weather Forecast on Exercism's Go Track. If you need help running the tests or submitting your code, check out `HELP.md`. If you get stuck on the exercise, check out `HINTS.md`, but try and solve it without using those first :)

Introduction

In the previous exercise, we saw that there are two ways to write comments in Go: single-line comments that are preceded by `//`, and multiline comment blocks that are wrapped with `/*` and `*/`.

Documentation comments

In Go, comments play an important role in documenting code. They are used by the `godoc` command, which extracts these comments to create documentation about Go packages. A documentation comment should be a complete sentence that starts with the name of the thing being described and ends with a period.

Comments should precede packages as well as exported identifiers, for example exported functions, methods, package variables, constants, and structs, which you will learn more about in the next exercises.

A package-level variable can look like this:

```
// TemperatureCelsius represents a certain temperature in degrees Celsius.
var TemperatureCelsius float64
```

Package comments

Package comments should be written directly before a package clause (`package x`) and begin with `Package x ...` like this:

```
// Package kelvin provides tools to convert
// temperatures to and from Kelvin.
package kelvin
```

Function comments

A function comment should be written directly before the function declaration. It should be a full sentence that starts with the function name. For example, an exported comment for the function `Calculate` should take the form `Calculate ...` . It should also explain what arguments the function takes, what it does with them, and what its return values mean, ending in a period):

```
// CelsiusFreezingTemp returns an integer value equal to the temperature at which water f
func CelsiusFreezingTemp() int {
        return 0
}
```

Instructions

Goblinocus is a country that takes its weather forecast very seriously. Since you are a renowned responsible and proficient developer, they asked you to write a program that can forecast the current weather condition of various cities in Goblinocus. You were busy at the time and asked one of your friends to do the job instead. After a while, the president of Goblinocus contacted you and said they do not understand your friend's code. When you check the code, you discover that your friend did not act as a responsible programmer and there are no comments in the code. You feel obligated to clarify the program so goblins can understand them as well.

1. Document package weather

Since goblins are not as smart as you are, they forgot what the package should do for them. Please write a comment for `package weather` that describes its contents. The package comment should introduce the package and provide information relevant to the package as a whole.

2. Document the CurrentCondition and CurrentLocation variables

The president of Goblinocus is a bit paranoid and fears uncommented variables are used to destroy their country. Please clarify the usage of the package variables `CurrentCondition` and `CurrentLocation` and put the president's mind at ease. This should tell any user of the package what information the variables store, and what they can do with it.

3. Document the Forecast() function

Goblinocus forecast operators want to know what the `Forecast()` function does (but do not tell them how it works, since unfortunately, they will get more confused). Please write a comment for this function that describes what the function does, but not how it does it.

Source

Created by

- @nikimanoledaki
- @micuffaro